iPhone Accessibility

Make your app compatible with VoiceOver

Session 121

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

iPhone Accessibility

Make your app compatible with VoiceOver

Chris Fleizach
VoiceOver Engineering

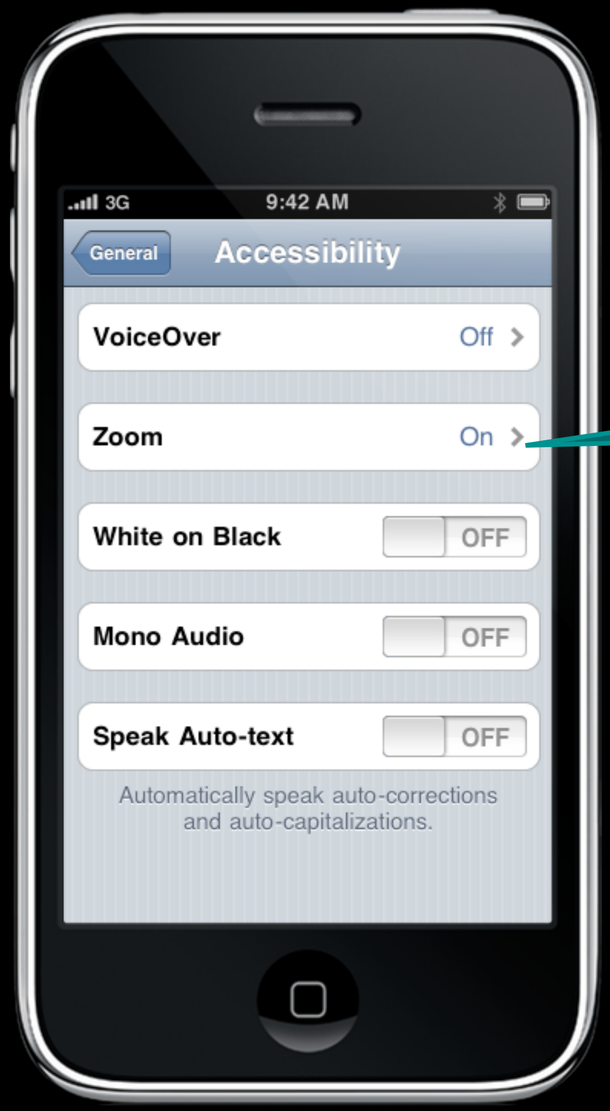
Eric Seymour
Manager, VoiceOver Engineering

Settings > General > Accessibility

Settings > General > Accessibility

VoiceOver
- Screen access software
- Speaks items on the screen
- Speaks activities and alerts
- Helpful sound effects
- “iPhone simple”
- “Safe touch”

Settings > General > Accessibility



Zoom

- Full-screen zoom
- Panning
- UI control panning
- Edge panning

Settings > General > Accessibility

White on Black

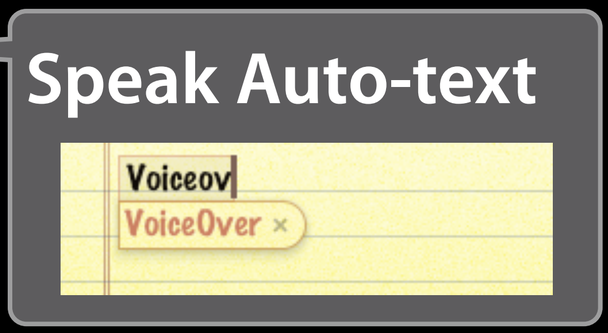
- Inverts the display colors

Settings > General > Accessibility

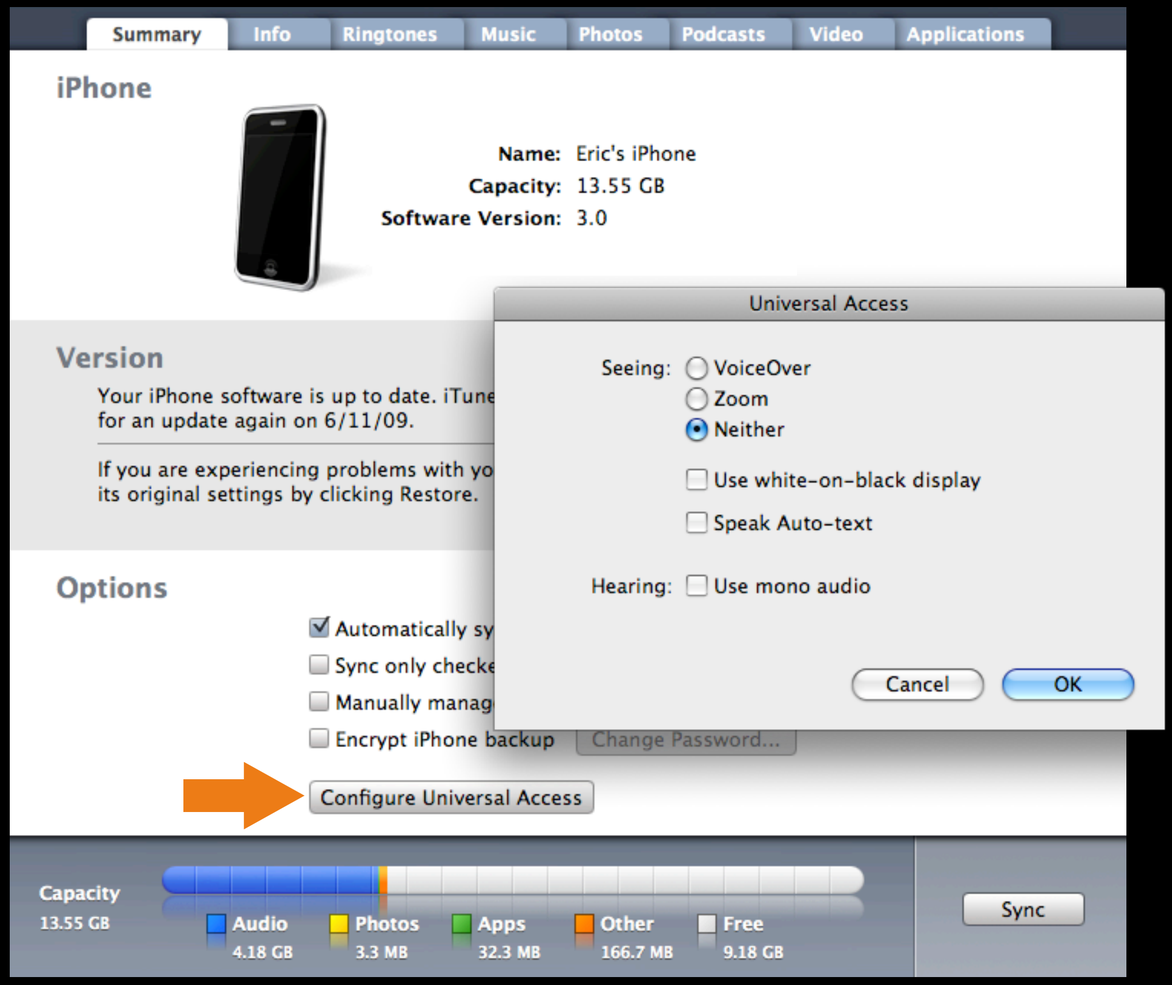
Mono Audio

- Both channels to both ears

Settings > General > Accessibility



Turning on Accessibility



Demo of iPhone Accessibility

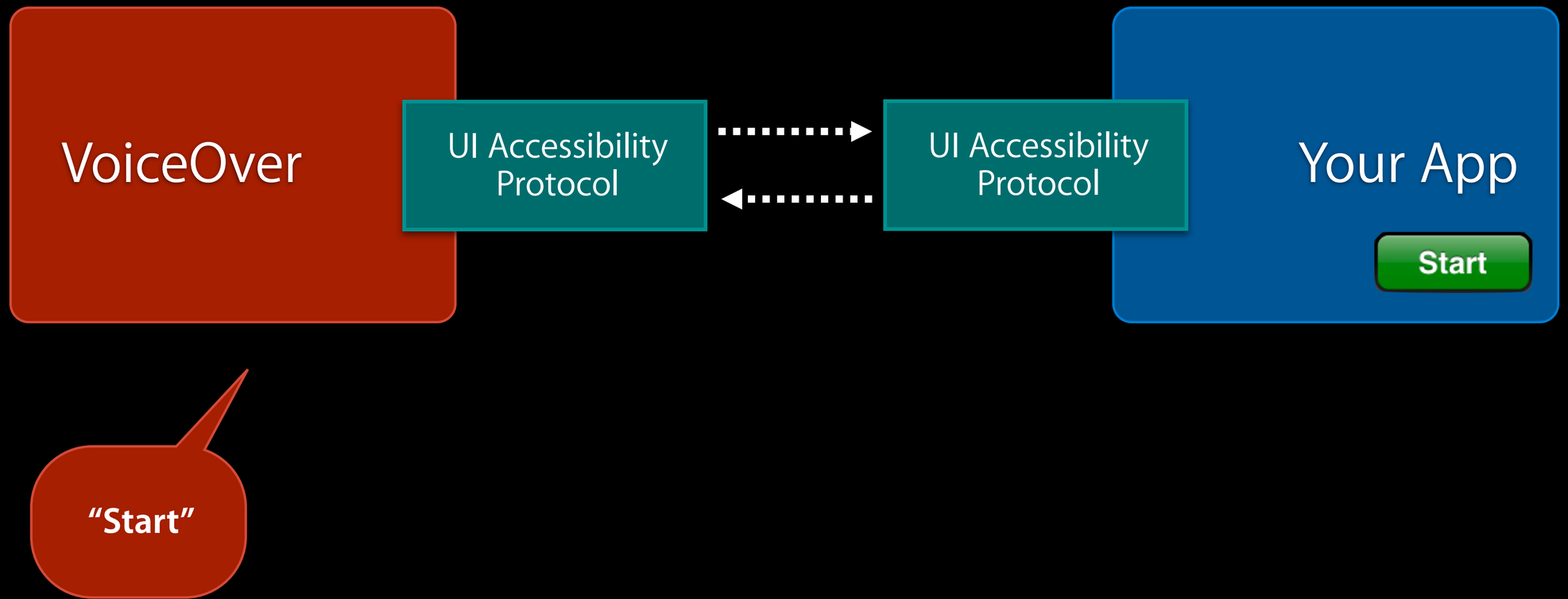
UI Accessibility API and Tools

Chris Fleizach
VoiceOver Engineering

Make Your App Accessible

- UI Accessibility API for iPhone 3.0 SDK New!
- Allows your app to provide UI information to an assistive application

Accessibility Architecture



Adding Accessibility to Your App

- The iPhone Accessibility API is designed for simplicity
- If your app uses standard UIKit controls, most of the work is done
- Many apps will only need to provide labels for UI elements that do not have associated text

Accessibility API: Attributes

- The Accessibility API consists of methods that return information about the element

```
- (BOOL)isAccessibilityElement
- (NSString *)accessibilityLabel
- (UIAccessibilityTraits)accessibilityTraits
- (CGRect)accessibilityFrame
- (NSString *)accessibilityHint
- (NSString *)accessibilityValue
```

Common Accessibility Attributes

- `- (BOOL)isAccessibilityElement`
  - Determines if the object is accessible (it's visible to VoiceOver)
- `- (NSString *)accessibilityLabel`
  - A textual representation of the element

Accessibility Traits

- `- (UIAccessibilityTraits)accessibilityTraits`
  - A set of traits that define status, state, and type

- Traits are OR'd together
- UIKit controls already return appropriate traits
- An element's traits should be OR'd with super

Accessibility Traits

Button	Link	Search Field	Image
Selected	Plays Sound	Keyboard Key	Static Text
Summary Element		Not Enabled	Updates Frequently

Other Accessibility Attributes

`-(CGRect)accessibilityFrame`

- The onscreen rectangle for the UI element

`-(NSString *)accessibilityHint`

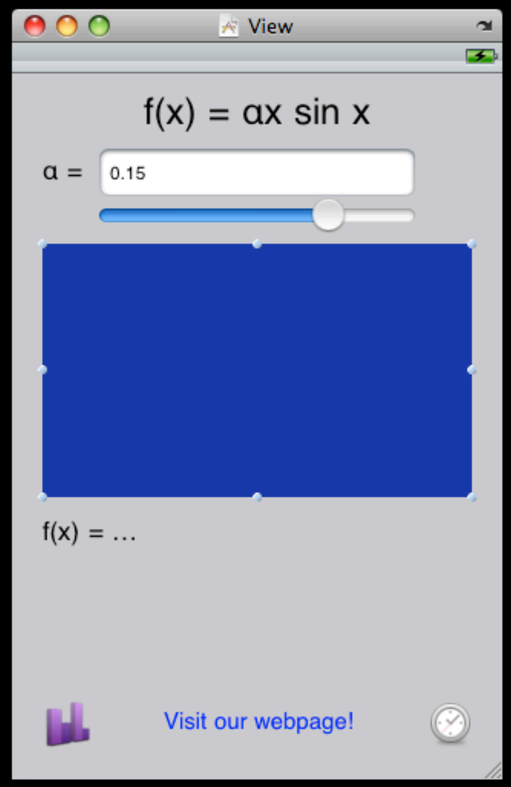
- A string that gives a hint for using the UI element

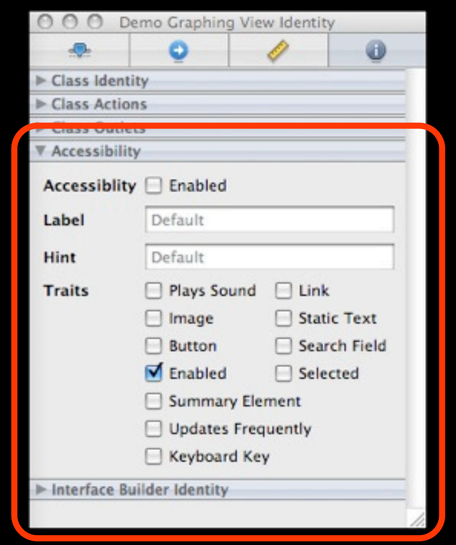
`-(NSString *)accessibilityValue`

- A dynamically changing value that describes the UI element

Adding Accessibility with Interface Builder

- Simple accessibility information can be added with Interface Builder

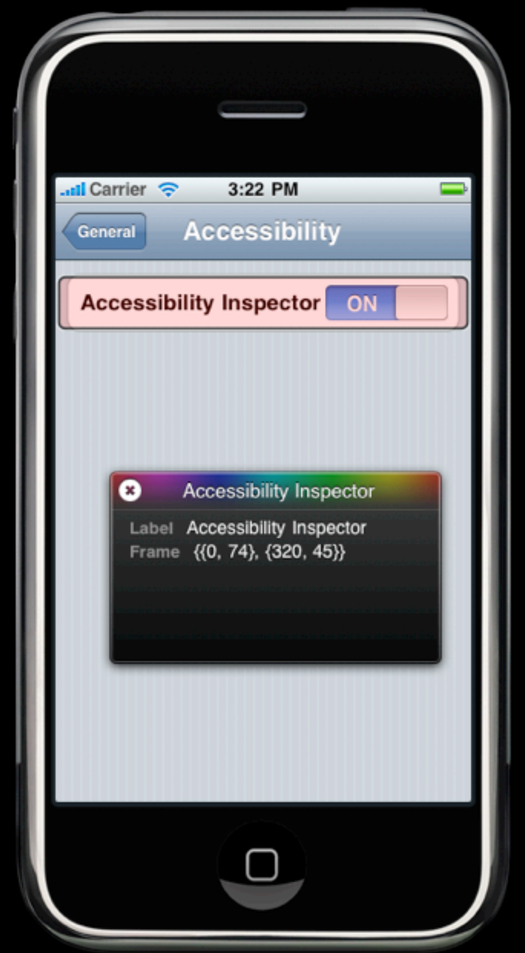




- Accessibility On/Off
- Label
- Hint
- Traits

Testing Accessibility

- Accessibility Inspector
- Settings > General > Accessibility



Adding and Testing Accessibility

Adding Accessibility Programmatically

- Accessibility can be added to programmatically created views
  - By setting accessibility attributes
  - By overriding accessibility methods

Adding Accessibility Programmatically

- If the accessibility information does not change over the course of the object's lifetime, you can use the accessibility setters

```
- (void)awakeFromNib {
    ...
    UIControl *control = [[UIControl alloc] initWithFrame:frame];

    [control setIsAccessibilityElement:YES];
    [control setAccessibilityLabel:@"Play"];

    [window addSubview:control];
    ...
}
```

Adding Accessibility Programmatically

- If the accessibility information of the object changes frequently, it may be appropriate to override methods

```
@implementation MyTemperatureView

- (BOOL)isAccessibilityElement {
    return YES;
}

- (NSString *)accessibilityLabel {
    return [_cityWeather currentTemperatureString];
}

- (UIAccessibilityTraits)accessibilityTraits {
    return [super accessibilityTraits] | UIAccessibilityTraitSummaryElement;
}

@end
```

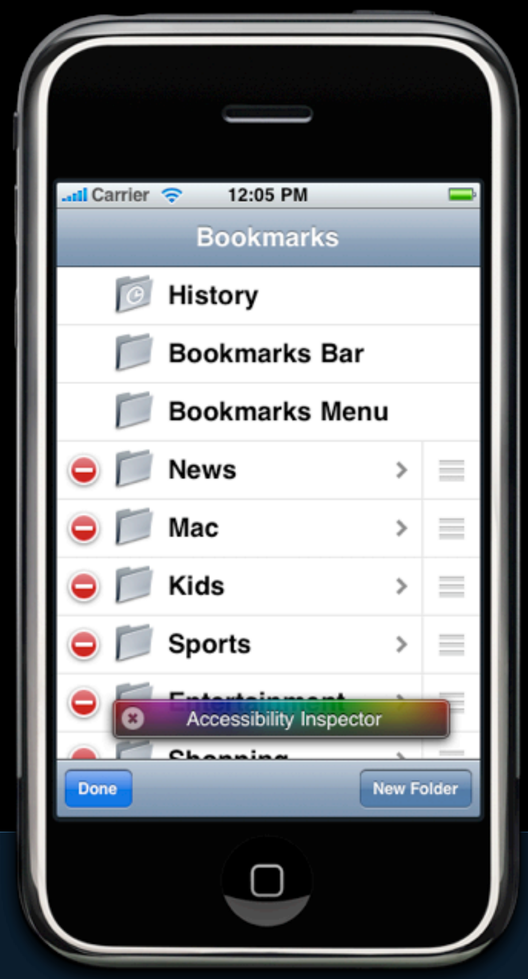
Notifications

- Screen change

  `UIAccessibilityNotificationScreenChanged`

- Layout change

  `UIAccessibilityNotificationLayoutChanged`




```
- (id)switchToBookmarks {
    ... // do necessary work to change views
    UIAccessibilityPostNotification(UIAccessibilityNotificationScreenChanged, nil);
}


- (id)beginEditingBookmarks {
    ... // do necessary work to change data
    UIAccessibilityPostNotification(UIAccessibilityNotificationLayoutChanged, nil);
}
```

Setting Accessibility Attributes

Accessibility Containers

- A UIView may contain disparate elements
- The Accessibility Container protocol allows a UIView to return accessible objects that represent “subelements”

Accessibility Containers

- A UIView that acts as a container should implement:

```
-(NSInteger)accessibilityElementCount
```

  - The number of elements the UIView contains

```
-(id)accessibilityElementAtIndex:(NSInteger)index
```

  - The element at the specified index

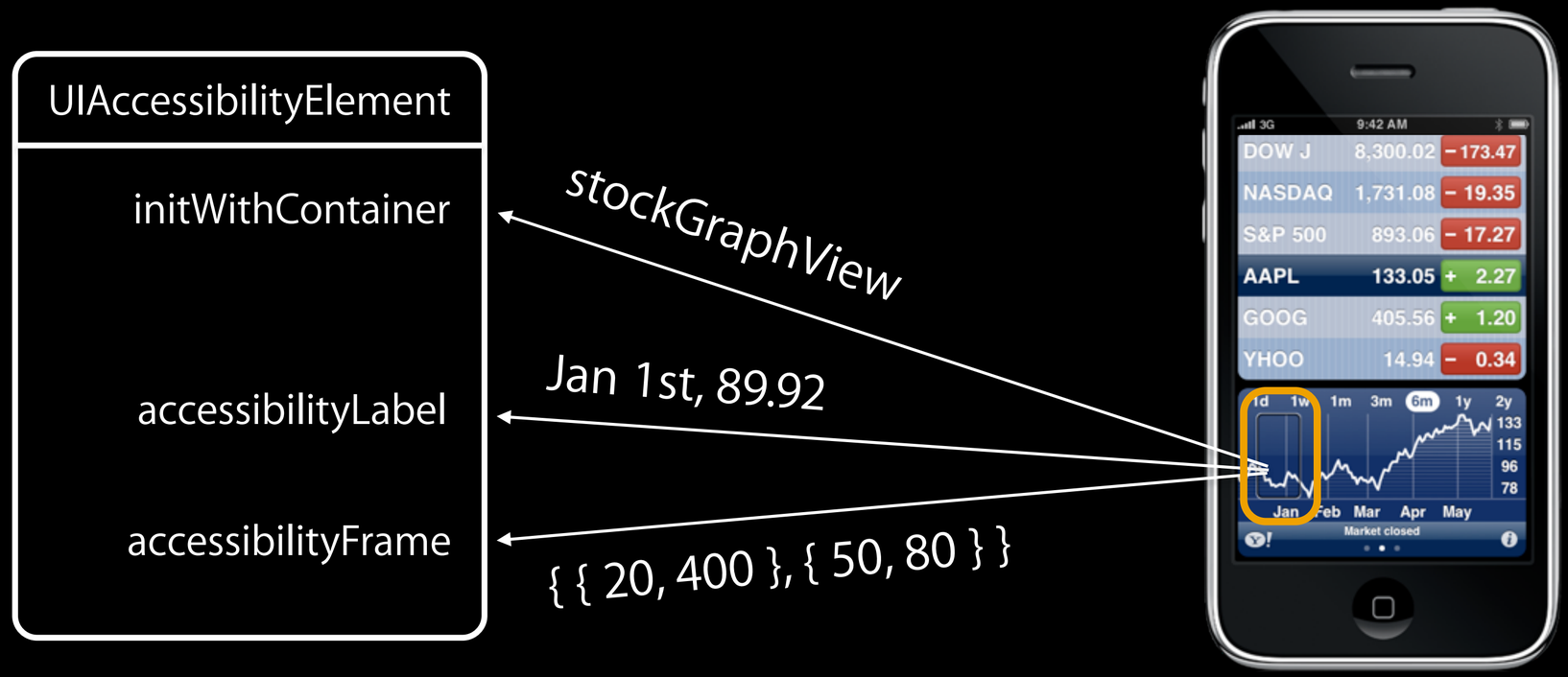
```
-(NSInteger)indexOfAccessibilityElement:(id)element
```

  - The location of the element in the list of elements

UIAccessibilityElement

- Elements in the container should be instances of UIAccessibilityElement
- UIAccessibilityElement encapsulates accessibility information

UIAccessibilityElement




```
UIAccessibilityElement *chartPart =
    [[UIAccessibilityElement alloc] initWithAccessibilityContainer:stockGraphView];
chartPart.accessibilityLabel = @"Jan 1st, 89.92";
chartPart.accessibilityFrame = CGRectMake(20, 400, 50, 80);
[_accessibilityElements addObject:chartPart];
```

Accessibility Container Protocol

Best Practices

- Use short, concise labels
  - ✅ Good: “Add city”
  - ❌ Bad: “Adds a city to the table view”

Best Practices

- Don’t include the type information in the label
  - ✅ Good: “Remove city”
  - ❌ Bad: “Remove city button”

Best Practices

- Hints should inform what happens when an element is activated—only if it is not clear

Good: “Plays a preview”

Bad: “Pressing here will play a preview of the current song”

Summary

- Accessibility expands the iPhone community
- Your apps are an important part of the accessibility story
- Adding accessibility is easy
- Use the Inspector and VoiceOver to test

Related Sessions

Enabling Your Mac Applications for Assistive Technologies (Mac OS X)	North Beach Wednesday 5:00PM
Improving Accessibility in Web Applications (Web)	Russian Hill Friday 9:00AM
Advanced Accessibility: Best Practices and New APIs in Snow Leopard	Mission Friday 2:00PM

Labs

Mac OS X Accessibility Lab	OS X Lab A Thursday 2:00PM
iPhone Accessibility Lab	iPhone Lab C Thursday 5:00PM
iPhone Accessibility Lab	iPhone Lab D Friday 2:00PM
Mac OS X Accessibility Lab	OS X Lab D Friday 2:00PM

More Information

Eric Hope
User Experience Evangelist
ehope@apple.com

Documentation

Accessibility Programming Guidelines for iPhone OS
http://developer.apple.com/wwdc/attendee/sessions/

VoiceOver User Manual
http://support.apple.com/manuals/iphone

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09